



Open Archive Toulouse Archive Ouverte

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible

This is a publisher's version published in:

<http://oatao.univ-toulouse.fr/22274>

Official URL

<https://doi.org/10.5220/0006540101430150>

To cite this version: Maignan, Sébastien and Bernon, Carole and Glize, Pierre *Speeding up the Search of a Global Dynamic Equilibrium from a Local Cooperative Decision*. (2018) In: 10th International Conference on Agents and Artificial Intelligence (ICAART 2018), 16 January 2018 - 18 January 2018 (Funchal, Portugal)

Any correspondence concerning this service should be sent to the repository administrator: tech-oatao@listes-diff.inp-toulouse.fr

Speeding up the Search of a Global Dynamic Equilibrium from a Local Cooperative Decision

Sébastien Maignan, Carole Bernon and Pierre Glize

IRIT, Université de Toulouse, Toulouse, France

Keywords: Multi-Agent System, Adaptation, Cooperation, Dynamic Equilibrium.

Abstract: Systems composed of many interdependent active entities working on shared resources can be challenging to regulate and multi-agent simulation is an efficient means for finding the suitable entities' behaviors. On the other hand, the search space for a stable solution of the system is usually forbidding and hinders any effort to solve the problem using a top-down approach. Furthermore, the complexity of possible global functions to optimize increases rapidly with the size of the system and can prove difficult to define and/or evaluate at each system simulation step. The difficulty when designing bottom-up systems is to be able to identify all their emergent properties and the parameters to modulate them. Here we propose a local cooperative decision making process that helps to stabilize such systems. These local processes prove to be very efficient to quickly find dynamic equilibrium solutions where the system continues to function and fulfills its global function. Regulation emerges from simple local interactions.

1 INTRODUCTION

As opposed to a stable equilibrium, a dynamic equilibrium is a balanced state which requires negative or positive feedbacks to stay in its steady state.

In order to remain functional or to operate in an efficient manner, complex dynamical or open systems often need to reach such a dynamic equilibrium. Economy (Aruoba et al., 2006), finance (Shimomura, 1998), ecology (Tuljapurkar and Semura, 1977) or biology (Bernado and Blackledge, 2010) are a few examples of areas in which such systems may exist.

As such, this study focuses on how a dynamic equilibrium may take place in cellular complex systems by modelling early stages of cell communities where interdependent cell species live together, but without having developed a communication system. Theoretically, a dynamic equilibrium should be reached for every cell to continue to exist and share resources. In principle, achieving such a state is possible if actions performed by the system's cells are specifically suited to give the feedbacks required.

A complex system is subject to unexpected behaviors due to bifurcations, non linearities, openness; and consequently cannot be designed in a top-down approach. A bottom-up design, where the global function of the system emerges from the interactions of its more basic components, becomes unavoidable and

makes knowledge of the suited actions impossible. Finding a way to constrain the actions locally made by the basic components for speeding up the convergence of the global system toward a dynamic equilibrium becomes therefore necessary.

The aim of this paper is thus to study how reaching this dynamic equilibrium can be sped up if local cooperation is used by the basic components.

Numerous works on cooperation found their origin on the computer tournament in the 1980s, based on Robert Axelrod's "The Evolution of Cooperation" (Axelrod, 1984). Usually the payoffs in game theory are fixed, which cannot be the case in the general situations such as the ones presented in this paper. The kind of cooperation we consider is seen as helping the most critical component in the system (Georgé et al., 2011), it is not simple cooperation towards a common goal (Hogg and Huberman, 1992) nor pure altruism; it is rather seen as benevolence.

For this study, multi-agent based simulation is used as it is known to be an efficient approach to simulate complex systems and study their properties (Moss, 2000). We consider a system where interdependent cells of different types are indirectly exchanging resources via a common environment. Exchanges are made using basic actions: a cell is able to gather resources from the environment, to internally transform some resources into others, and to

release resources in the environment. The studied system is simulated as a Multi-Agent System (MAS) (Wooldridge, 2009) where cells are agents situated on nodes of a grid that form their environment. These cells are not aware of their neighbors' actions, are not able to directly communicate with them, and are only able to act for gathering, producing or transforming resources units. A cell-agent may live a certain amount of time depending on the current resources it possesses and should it die, its node is in charge of finding a replacement. Once released by a cell-agent at a particular node of the environment, a resource may circulate and be gathered by another cell-agent on its own node. The environment also produces resources and regenerates the cells when they disappear; and, in this sense, nodes are also considered as agents because they take decisions for processing this regeneration.

The aim is therefore to endow agents with purely local behaviors based on cooperation in order to study how the simulated system behaves globally and whether these cooperative behaviors enable finding a dynamic equilibrium in an efficient manner. In other terms, we want to verify whether cooperation, as defined here, is an effective behavior to build an agent-based simulation in a bottom-up way considering that agents, as in real life, have a very limited knowledge of their surrounding and no knowledge at all of the global state to reach.

The paper is structured as follows. The features of the studied system as well as the cooperation used for replacing component-agents and selecting their actions are presented in section 2. Section 3 presents some results when cooperation is used in systems with populations of 2 or 10 component species. Before concluding in section 5, section 4 studies the stability and robustness of the cooperative heuristics.

2 SIMULATED SYSTEM

As mentioned earlier, the studied system simulates early stages of cell communities where interdependences exist between cell types but no communication system exists yet to coordinate their actions. While being a very simplified version of actual cell communities, this system is already complex enough to observe emergence of some interesting behaviors.

Cells are the interdependent components of the system, they release into or gather, from the environment, molecules representing resources which they, as a whole, require to survive. Reaching a dynamic equilibrium in such a cellular system means that all the cells find a way to survive by exchanging the right

resources in the right way without having any prior knowledge about this. The system template is loosely based on cell communities sustaining each other by producing specific resources required by other cell types and receiving vital resources from other members of the community. The working model of each cell is a very simplified version of actual cells and only gathering, release and production functions have been kept.

This section presents the general features of the simulated system and describes the cooperative behaviors used by its agents in order to reach a dynamic equilibrium in a faster way. The parameters actually used for the simulation are given in section 3.

2.1 System

The simulated system has therefore two parts: cell agents (later on, simply "cells") and their environment. This environment is a two-dimensional torus seen as a grid of node agents (later on, simply "nodes") on which cells are regularly placed.

A given number of resources (nR) are available in the system. Each node of the environment contains external resources eR_i where i is the resource index. Each cell contains internal resources iR_i and a set of actions. Three different types of actions enable a cell to act on resources:

- Gathering: Transfers a given amount from the node resource eR_i to the cell resource iR_i .
- Production: Transforms a given amount of linear combination of iR_i s into another kind iR_j i.e. $\sum_i a_i \cdot iR_i \rightarrow iR_j$ where $\sum a_i = 1$.
- Release: Transfers a given amount of cell resource iR_i to the node resource eR_i .

Only one action can be performed by a cell during each simulation cycle. Performing this action will cost the cell a certain amount of one particular resource that can be viewed as its energy source. However, as in real life, a cell does not have any knowledge that a particular resource is actually its energy source and is therefore unable to optimize its behavior towards energy maximization.

The longer a real cell exists the more errors it will accumulate. These errors are simulated by giving each cell a probability to disappear which increases with its age. When a cell fails, it is immediately replaced on the node it occupied (see section 2.4).

Internal or external resources are divided into two subsets: $SetA = \{R_i, i \leq \frac{nR}{2}\}$, which contains abundant resources supplied by the nodes, and $SetB = \{R_i, i > \frac{nR}{2}\}$, which contains resources only produced

by cells. Resources from *SetA* are replenished every cycle either at each node and are always available to cells. On the contrary, a constant amount of resources from *SetB* is removed every cycle from every node to simulate their transport away from the system. Resources diffuse from high content nodes to low content neighboring nodes¹. To summarize, there are three mechanisms to modify resource content at each nodes: passive diffusion, active gathering/release from cells and outside world influence (positive for *SetA* and negative for *SetB*).

The overall goal of the system is to reach a dynamic equilibrium and to provide a steady output of resources from *SetB*. Robustness to resources changes and/or component distribution is also an important criterion to consider.

As introduced before, cooperation is used to regulate the actions performed by the cell/node agents in the system, and it is supposed that every agent knows that its neighbors expect reciprocal cooperation. A cooperative behavior is a balance between selfishness and altruism and is implemented in two parts: (1) during the action selection for each cell (see section 2.3), and (2) during the creation of a new cell (see section 2.4).

2.2 Resources Criticality

The notion of criticality is used by a cooperative agent to decide which agent is considered as having the highest degree of dissatisfaction in its neighborhood (Noël and Zambonelli, 2015) in order to help it if possible.

Here, cells do not know how other cells perform so their degree of dissatisfaction can only be related to resources variations. Thus internal and external resources criticalities reflect this information. In the same way, nodes also require access to external resource criticality to operate the cooperative agent replacement process (see section 2.4).

Resource criticality is a measure of the current available quantity of a resource combined with its forecast availability. The current available quantity of a resource could translate as “is it rare or plentiful?”. Its forecast availability could translate as “is it disappearing from the environment or not, and how fast?”. So, an abundant resource replenished every cycle will have a very low criticality whereas a very rare resource consumed by all neighbor cells will have a very high criticality. Whatever the actual numerical

¹The fact that cells are immobile does not reduce the problem scope since resources may move and each agents neighborhood changes over time.

implementation of the criticality evaluation, its fundamental characteristic is that it can be compared and ranked.

Criticality is a local measure available only to a single cell and its containing node.

2.3 Cooperative Action Selection Process

Every cycle, each cell selects an action and performs it.

The default algorithm, for comparison purposes, consists in randomly choosing this action. The action selection process proposed is based on cooperation between cells to improve the status of resources in critical situations.

Firstly, resources criticalities for all resources, internal and external, are calculated and ranked.

All actions that could be performed by the cell are then put in a working set. For example, a gathering action will only be added if the external resource this action involves is available in the environment.

Starting from the most critical resource, actions from the working set are sorted according to their positive impact on this resource i.e. the amount of resource it would add either internally or externally. If several actions have the same impact, the one with the least impact on other resources is selected. If no action can improve the status of this critical resource, the next most critical resource is evaluated. If no action can improve the status of any critical resource, a random action is selected.

The final action selected after this process is then performed by the cell.

This process will select all the necessary actions to gather, produce and release all the resources ensuring the cell’s survival and that of its neighbourhood.

Several experiments (data are not shown here) proved that a random selection of actions always quickly leads to a global system failure. In consequence, all the results presented thereafter use the cooperative process for action selection.

2.4 Cooperative Agent Replacement Process

Any given cell can fail for two reasons: it has become too old or its energy resource has fallen to zero. A failed cell disappears and needs to be replaced so that each node of the system is occupied at all time. The new cell has to be the daughter of an existing cell from its neighbourhood.

All cells in a specific radius (small compared to the system size) around the failed cell position are

candidates to replace it, but only those with enough energy resource to last a predefined number of cycles are considered as potential candidates.

The default algorithm, applied by a node to replace the cell which failed at its location, consists of randomly selecting one of the potential candidates. The agent replacement process proposed uses cooperation to make the node select the candidate that will help the most critical external resource in its direct neighborhood.

Only external resources are used during this process since the nodes and cells do not have any knowledge of the resource content in the neighboring cells. A node has access to resource criticalities (as defined in section 2.2) in the neighboring nodes. This is the way to make cooperation possible.

First, for each resource eR_i , its criticalities on each neighboring nodes are calculated and compared, and the worst one becomes the criticality for eR_i .

Then, starting from the most critical eR_i , candidate cells are ranked according to the amount of iR_i they released in the recent past. If more than one candidate released iR_i , they are ranked by the amount released and the top half remain candidates for the next comparison round using the next most critical resource. This is repeated until only one candidate remains or all critical resources have been tested. The right candidate is then duplicated (i.e. with its current sets of resources and actions) to the failed cell position.

If there are no candidates, the node stays empty and the process starts over during the next cycle. This can potentially lead to full system halt if mass cell failure occurs.

3 SIMULATION RESULTS

In order to evaluate the efficiency of local cooperation – as described in the previous section – to regulate a running system, several simulations were performed and are described in this section. After describing how the simulation is initialized, results are given for simulations where 2 or 10 interdependent cell types are considered.

3.1 System Initialization

“Abundant” resources from $SetA$ are distributed randomly on the nodes of the system and within the cells. “Rare” resources from $SetB$ are distributed randomly within the cells to provide them with enough energy resources to survive for at least a given number ($nMinS$) of cycles.

Initially, nCa cell types are distributed on the system nodes; these types are classes of cells that differ by their energy resources and their production action.

For each cell type $j \in \{1, \dots, nCa\}$, one resource from $SetB$ is chosen as the energy source for this type (which is not known by the cells). Furthermore, this cell type is given a set of actions: shared common actions to gather or release any type of resource, along with one specific production action allowing this cell type to produce one unit of its energy resource using a_{j1} units of one resource from $SetA$ and a_{j2} units of one resource from $SetB$:

$$a_{j1} \cdot iR_j + a_{j2} \cdot iR_{(j+1+\frac{nR}{2}) \bmod nCa} \rightarrow iR_{(j+\frac{nR}{2})} \text{ with } a_{j1} + a_{j2} = 1$$

At the start of the simulation, each cell instance of a given type j has a unique random set of (a_{j1}, a_{j2}) .

To illustrate this a 2-cell type and 4-resource system would give the following cyclic interdependence:

$$\text{cell type 1: } a_{11} \cdot iR_1 + a_{12} \cdot iR_4 \rightarrow iR_3$$

$$\text{cell type 2: } a_{21} \cdot iR_2 + a_{22} \cdot iR_3 \rightarrow iR_4$$

In other words, each cell is able to produce its energy resource but needs resources only produced by another cell type to do so. Without any knowledge about its energy resource, and although it is vital to its survival, a cell must somehow manage to release some of it so that other cell types can survive.

The proportion and location of cells using a specific energy resource can be set randomly or following a predefined pattern. In all the following examples, cells of a given type are spatially grouped together at the beginning of the simulation and the proportions for each cell type are the same.

For the results given hereinafter, the features of the simulated system are as follows: nCa is set to 2 or 10, nR is set to 20 i.e. $SetA = \{R_0 \dots R_9\}$ and $SetB = \{R_{10} \dots R_{19}\}$ and $nMinS$ is arbitrarily set to 80.

The size of the simulated system is 60×60 with $\frac{3600}{nCa}$ of the cells for each type. The speed at which resources can move from one node to its neighbors (diffusion factor) is set to 0.01 unit/cycle. The radius for candidate selection is set to 4 nodes in the cooperative agent replacement process: this is larger than the direct neighborhood of a node using a radius of 1 (involving 8 nodes) but lower enough for not considering the whole grid as its neighborhood.

A rough estimation of the parameter space for the system is as follows: every cycle, each of the 3600 cells may perform 1 out of 40 actions that are common to all cell types, and 1 action, specific to its type. Considering that cell type populations are roughly equivalent, there are $(40 + nCa)^{3600}$ possible choices every cycle. As observed in several simulations, the system can be balanced within 1000 cycles.

The parameter space for this period is therefore at least $(40 + nCa)^{3600^{1000}}$. Random walk in such a huge space is very unlikely to succeed. Actually, starting from an identical initial state, in all the experiments performed none of the system configurations of the random simulation ever matched one of the cooperative simulation.

We are fully aware that comparing the results obtained by the system, when cooperation is used to constrain the actions performed locally by the agents, with other works in the same vein is compelling. However, to the best of our knowledge, we did not find simulations of similar systems to perform such a comparison. Furthermore, since a global cost function cannot be considered to use machine learning approaches such as neural networks (Colak, 2006) or genetic algorithms (Gen and Cheng, 2000), we chose to randomly explore the search space to show the difficulty to find solutions and their sparsity.

3.2 Two Interdependent Cell Types

For this scenario, there are $nCa = 2$ cell types.

The cooperative system quickly reaches a dynamic equilibrium (figure 1) whereas random selection of new cells cannot maintain a dynamic equilibrium after the stabilization period (figure 2).

In figure 1, resources R_{12} to R_{19} quickly disappear from the system since none of the two cell types can produce or use them. These resources are then always the most critical ones but no action can improve their state. This is not a problem for the system as a whole as these resources do not impact the viability of the cells.

It is noteworthy to mention that the spatial distribution of type 1 and type 2 cells is not regular and does not display any kind of (visually) distinctive pattern (result not shown here).

Interestingly after a few thousand cycles, only one instance of type 1 and one instance of type 2 cells are left from the initial random sets and occupy all the simulation space. It can be noted that the surviving instances selected by the cooperation process always have a_{j1} above 0.8 and a_{j2} below 0.2. This can be explained by the fact that cells using more scarce resources from *SetB* will have a shorter lifespan and be replaced more often. This indicates that the system maximizes the use of abundant resources (R_1 to R_9) and minimizes the consumption of rare resources (R_{10} and R_{11}).

3.3 Ten Interdependent Cell Types

For this scenario, there are $nCa = 10$ cell types.

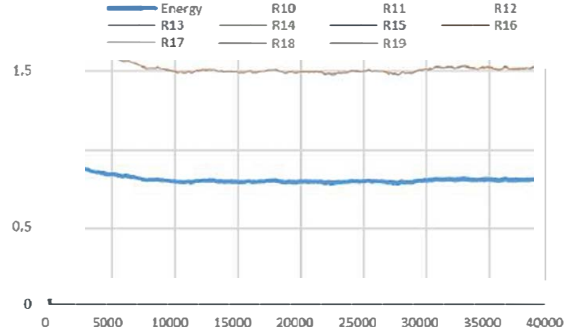


Figure 1: Result of dynamic equilibrium between 2 interdependent cell types using cooperation. The x axis represents the number of simulation cycles. The mean quantity of each resource from *SetB* ($R_{10} - R_{19}$) in the nodes (top curve) and the mean energy of the cells (bottom thicker curve) are represented on the y axis. R_{10} and R_{11} are nearly superimposed and $R_{12} - R_{19}$ are equal to zero very early in the simulation.

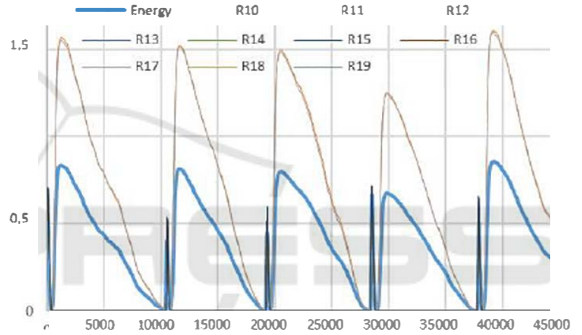


Figure 2: Same setup as Fig. 1 but using random selection for failed cells. Each spike denotes a system restart when at least one component type disappeared compromising the future of the whole system.

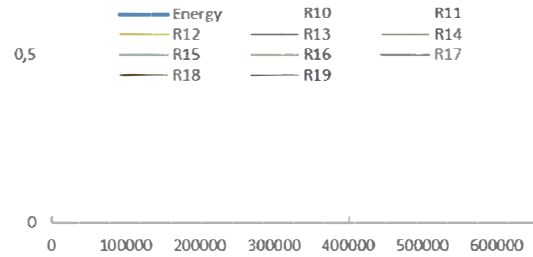


Figure 3: Result of dynamic equilibrium between 10 interdependent cell types using cooperation. Resources are nearly surimposed and show very small fluctuations (bottom curve).

As shown in figure 3, after around twenty thousand cycles the system reaches a dynamic equilibrium using the cooperation selection process (contrary to figure 4 where the random one is used). All resources from *SetB* are used by one cell type or another. This explains why none of the resource quantities falls to

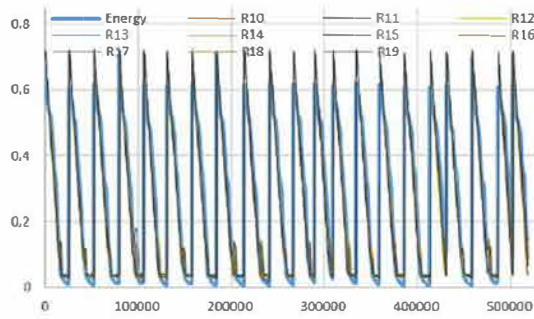


Figure 4: Same setup as figure 3 but using random selection for failed cells. The number of cycles before failure may vary but the system is always unstable and never reaches a dynamic equilibrium state (noted also in all the different random simulations launched).

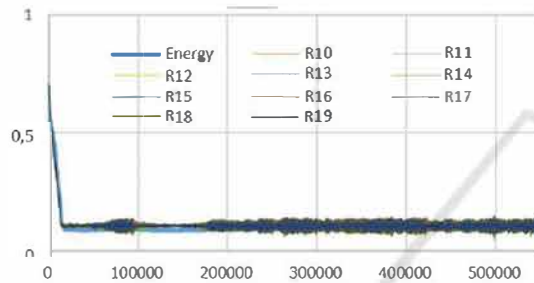


Figure 5: Same setup as figure 3 but using a slower resource diffusion factor ($10\times$). This impacts the resource values and the cell type distribution. The fluctuations are cyclic instead of random.

zero.

Again, when the system becomes stable, it is filled with instances for which a_{j1} is above 0.8 and a_{j2} is below 0.2.

Using the same system setup but with a much slower diffusion factor (10 times slower) the system behaves slightly differently (see figures 5 and 6). Fluctuations of a resource R_j become cyclic and correlated with those of R_{j-1} and R_{j+1} . This correlation is the result of resource coupling in the production actions of the interdependent cell types. The diffusion factor impact can be explained by the fact that cells will deplete the medium of resources they need before they can be replenished by the passive diffusion mechanism. Cells will start to die but resource quantities will allow a small population to subsist. Eventually diffusion will replenish the depleted resources and cells will thrive again.

In parallel, cell type populations also present cyclic and correlated behaviors (these data are not shown). Although in most simulations the system is dynamically stable, it happened on some occasions that the fluctuation amplitudes increased with time and led to system failure.

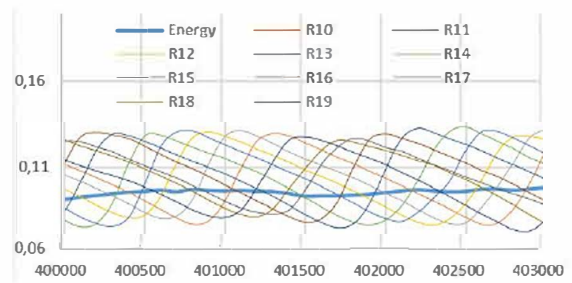


Figure 6: Same setup as figure 5 showing the resource oscillations and interdependence. Cell type population displays a similar oscillating behavior.

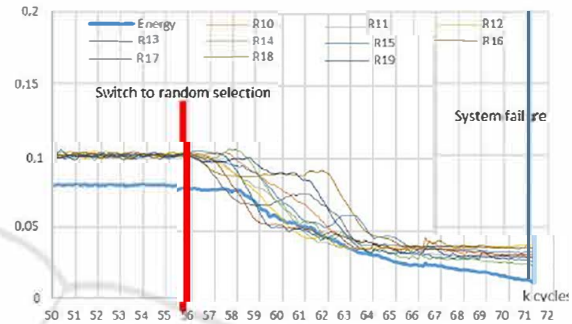


Figure 7: When a stabilized system is switched to random selection of new cells, it oscillates for a while before eventually failing after around 15,000 cycles (when one cell type vanishes from the system). For clarity, cycles 0-50,000 were removed.

4 COOPERATION AND DYNAMIC EQUILIBRIUM

Through three experiments derived from the simulation used in section 3.3, this section studies how robust is cooperation when a certain percentage of cells are replaced using the basic random algorithm.

4.1 Solution Stability without Cooperative Selection

Using the scenario and solution found in section 3.3 with 10 cell types, the system is switched from a cooperative selection of new cells to a random one.

At the time of the switch, $\{(a_{j1}, a_{j2})\}$ for the production actions of all cells has been optimized by the cooperation for the best possible performance. The consequence is that the system is able to survive for much longer than a new system using the much wider selection of $\{(a_{j1}, a_{j2})\}$ available at the simulation start. Nevertheless, as illustrated in figure 7, after several thousand cycles the equilibrium between cell types is not maintained and the system fails.

4.2 Robustness to Noise

This scenario studies the robustness to noise of the cooperative algorithm described in section 3.3. Each time a cell needs to be replaced there is a given probability to use the random selection or the cooperative process.

An insight of the effect of cooperation is given in figure 8. The cooperative process is efficient in stabilizing the system even when new cells are selected randomly 99% of the time. Nevertheless, at high random percentage the system is more sensitive to initial conditions and cell distribution. Statistical studies need to be performed in order to evaluate the exact percentage of cooperative replacement process required to hold the system steady.

4.3 Saving a Failing System

In this experiment, a system, as described in section 3.3, is simulated using random selection of cells until failure. Every thousand cycles the system is saved. Failure occurs after around 23 thousand cycles. Then every saved instance of the system is continued using cooperative selection of cells and the system survival time is measured. This procedure is not equivalent to restarting the system with different random seeds since during system restart each cell receives enough energy resource to survive a given number of cycles.

As shown in figure 9 and 10, the system can be saved if cell selection is switched to cooperation early enough (around 15K cycles). After this time, it is not possible to rebalance the system. One reason is that some cell type populations are so low that they cannot be increased before random failure from old age

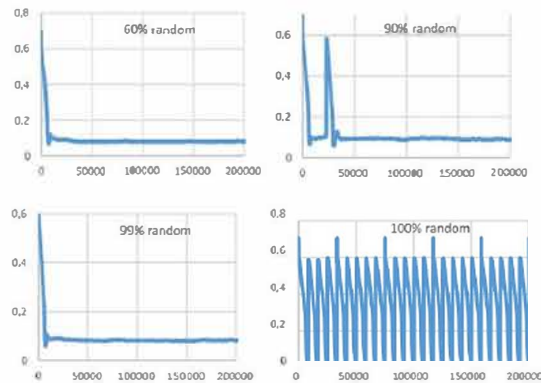


Figure 8: These graphs represent the mean cells energy during the simulation when replacement cells are selected increasingly randomly. Up to 99% of random selection the system is still stable. Nevertheless, some of the random starting system configurations cannot be balanced as demonstrated by the restart pikes observed in most of the graphs.

or low energy remove them from the system. Another reason is that the number of instances with interesting $\{(a_{j1}, a_{j2})\}$ for the system balancing might not be present anymore.

4.4 Scalability

The largest system tested so far consists of a 300×300 grid with 30 cellular types and 60 different resources. After 5K cycles, the system reaches a dynamic equilibrium and is still stable after over 62K cycles. Nevertheless due to its size this experiment has not been repeated extensively and is only mentioned here as an indicator of the scalability for this approach.

5 DISCUSSION

This article aimed at showing that a specific kind of cooperation enables a global dynamic equilibrium to be found in a faster way than randomly and has used a multi-agent system to simulate a complex system of interdependent components where the dynamic equilibrium means survival of the components.

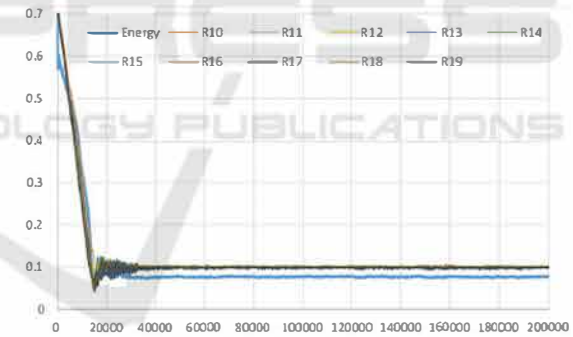


Figure 9: The system was simulated for 15K cycles using random selection then switched to cooperative selection. The system is stabilized and reaches dynamic equilibrium.

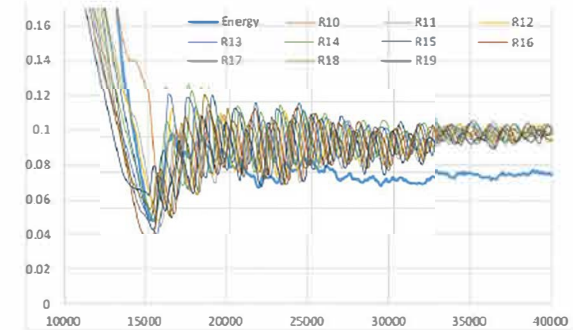


Figure 10: Close up on the transition phase from figure 9. The system oscillates for around 20K cycles before cell type populations and resources are balanced.

Optimizing localized production and consumption of various resources globally can prove to be a very challenging endeavor. The formal global function to minimize is not straightforward to derive from the system description nor to evaluate at each cycle. Therefore it becomes necessary to endow components with a local attitude which enables emergence of the right global behavior. The local actions considered here always tend to help the most critical entity in the local environment of the component, as such they are said to be cooperative.

A cellular system in which resources are produced or consumed by cells was chosen to perform experiments with a growing number of cell types, and related results were presented.

As expected from its huge size, a random exploration of the search space is unsuccessful even when repeatedly tried with different starting systems (these data are not shown). On the other hand, local cooperation allows components/cells to regulate efficiently the resource management although the global goal of the system is locally unknown and the cell knowledge about its neighborhood is quite restricted.

Furthermore, the processing is totally distributed within the nodes and cells. Their interactions depend only on a very limited neighborhood: the ones to select a new cell to duplicate. This makes the simulation very simple to parallelize and reduces computation time. Also the code of the nodes and cells which facilitates the progression in the search space is extremely simple. This stems from the fact that the local decision based on cooperation replaces the complexity of the global problem.

The result of the resolution process is totally emergent in the sense that no part of the code is guided by an evaluation of the overall quality of the solution in progress (which is almost impossible to estimate).

The more resources to regulate the more challenging the regulation is. In this paper we have demonstrated that local cooperative processes to select cell actions to perform and to replace failed components are able to regulate a system with 10 interdependent cell types working on 20 resources.

Two interesting observations have been made during these simulations. First, the factors in the production actions $\{(a_{j1}, a_{j2})\}$ always tend to be large for abundant resources from *SetA* and small for scarce resources from *SetB*. This is intuitively a good thing to do when you optimize such a system. Nevertheless, this is not coded in the cooperative processes but emerges from them.

Secondly, the mean value of the resources quantities in the system tends toward a value that is not a parameter in the system simulation: around 0.8 for 2

cell types, and 0.1 for 10 cell types. These values also emerge from the cooperative processes and no single parameter in the simulation seemed to be able to alter them significantly (these data are not shown).

As demonstrated in section 4.1, stabilization of the system with the cooperative processes must be continuous since it eventually fails sooner or later if switched to random selection. Nevertheless, as shown in section 4.2, only a very small amount of cooperative selection is necessary to continue to stabilize the system.

REFERENCES

- Aruoba, S. B., Fernandez-Villaverde, J., and Rubio-Ramirez, J. F. (2006). Comparing solution methods for dynamic equilibrium economies. *Journal of Economic dynamics and Control*, 30(12):2477–2508.
- Axelrod, R. (1984). *The Evolution of Cooperation*. Basic books.
- Bernado, P. and Blackledge, M. (2010). Proteins in dynamic equilibrium. *Nature*, 468(7327):1046–1047.
- Colak, S. (2006). *Neural Networks Based Metaheuristics for Solving Optimization Problems*. University of Florida.
- Gen, M. and Cheng, R. (2000). *Genetic algorithms and engineering optimization*, volume 7. John Wiley & Sons.
- Georgé, J.-P., Gleizes, M.-P., and Camps, V. (2011). Cooperation. In Di Marzo Serugendo, G., Gleizes, M.-P., and Karageorgos, A., editors, *Self-organising Software: From Natural to Artificial Adaptation*, pages 193–226. Springer.
- Hogg, T. and Huberman, B. A. (1992). Better than the best: The power of cooperation. *SFI*, pages 163–184.
- Moss, S. (2000). Editorial introduction: Messy systems—the target for multi agent based simulation. In *Multi-Agent-Based Simulation*, pages 1–14. Springer.
- Noël, V. and Zambonelli, F. (2015). Methodological guidelines for engineering self-organization and emergence. In *Software Engineering for Collective Autonomic Systems*, pages 355–378. Springer.
- Shimomura, K. (1998). A dynamic equilibrium model of durable goods monopoly. *Journal of Economic Behavior & Organization*, 33(3-4):507–520.
- Tuljapurkar, S. and Semura, J. (1977). Dynamic equilibrium under periodic perturbations in simple ecosystem models. *Journal of Theoretical Biology*, 66(2):327–343.
- Wooldridge, M. (2009). *An Introduction to Multiagent Systems*. John Wiley & Sons.